

Robot Wars Registration System

project outline

Rai Faizan Qasim | OOP | FA23-BSE-154

**Project Goal:**

Develop a robust and user-friendly Java application with a graphical user interface (GUI) using JavaFX that allows users to register and manage their participation in a Robot Wars competition.

**Technologies:**

* Java
* JavaFX

**Object-Oriented Programming (OOP) Concepts:**

* Encapsulation
* Inheritance
* Polymorphism
* Abstraction
* Exception Handling

**Project Phases:**

**1. System Design and Object-Oriented Modeling**

* Define user types (Participant, Viewer) with clear distinctions in functionalities.
* Design a comprehensive class structure that leverages OOP principles:
  + User (encapsulates email, password)
  + Participant (inherits from User, adds team name, member names, robot weight, robot name)
  + Robot (abstract class, defines weight category based on weight)
    - HeavyweightRobot (inherits from Robot)
    - LightweightRobot (inherits from Robot)
  + Registration (handles user registration and login)
  + Competition (abstracts participant management, can be extended for future functionalities like scheduling battles)

**2. User-Friendly Interface with JavaFX**

* Design an intuitive and visually appealing GUI using JavaFX components:
  + Login screen (email, password fields, login button) with clear error messages for invalid credentials.
  + Registration screen (email, password confirmation fields, registration button) with appropriate input validation.
  + Participant registration screen (team name, member names, robot weight, robot name fields) with user-friendly guidance.
  + Confirmation screen (displays registered information) for user verification.
  + Main menu (options for participant and viewer functionalities) with clear labeling.

**3. Core Functionality Implementation**

* **Secure User Management:**
  + Implement user registration with robust email and password validation (including password strength checks).
  + Store user credentials securely using hashing techniques to prevent unauthorized access.
  + Develop login functionality with credential verification and exception handling for invalid login attempts.
* **Participant Registration and Data Management:**
  + Allow participants to register their team and robot information with clear instructions.
  + Implement logic to determine robot weight category (Heavyweight/Lightweight) based on weight input and provide informative messages.
  + Use arrays to store participant data (team name, member names, robot details) efficiently.
* **Persistent Data Storage:**
  + Save participant data in a text file upon registration confirmation for data persistence.
  + Implement methods to load data from the text file on program startup to maintain user information.
* **OOP in Action:**
  + Encapsulate data within classes (private fields, public getters/setters) for data protection and access control.
  + Utilize inheritance for user types and robot weight categories to promote code reusability.
  + Implement polymorphism for different robot types (e.g., display robot information in a consistent format).
  + Abstract the competition functionality for potential future enhancements (e.g., scheduling battles) using abstract classes.
  + Implement exception handling for potential errors (e.g., invalid data input, file operation issues) to make the program more robust.

**4. Testing and Deployment - Ensuring a Smooth Run**

* Write unit tests to thoroughly test core functionalities (registration, login, data storage, etc.) for a reliable application.
* Fix any bugs you find in the code to ensure a smooth user experience.
* Package the application for deployment, making it ready to run on other computers for wider accessibility.

**5. Extra Feature:**

* **Enhanced Viewer Experience:**
  + Implement functionalities for viewers (e.g., view a list of participants or robot details) to provide more engagement.